



NARSIMHA REDDY
ENGINEERING COLLEGE

◆ (AUTONOMOUS) ◆

JAVA PROJECTS

◆ A COMPLETE PROJECT GUIDE ◆



CORE JAVA



OOPs
CONCEPTS



DATA
STRUCTURES



JDBC &
DATABASE



EXCEPTION
HANDLING



FILE
HANDLING



MULTITHREADING



REAL TIME
PROJECTS



BEST
PRACTICES



P SURYA GANESH

◆ ASSISTANT PROFESSOR ◆

DEPARTMENT OF

AIML

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)



www.nrcmec.org



principal@nrcmec.org



Kompally, Hyderabad, Telangana - 500100

S.NO	PROJECT TITLE	PAGE NO
1	Smart Drone Delivery Management System	2-3
2	Digital Wallet & UPI Payment Simulator	4-5
3	Smart Campus Event Management Portal	6-7
4	Smart Carbon Footprint Tracking System	8-9
5	Disaster Alert & Emergency Response System	10-11
6	EV Charging Station Booking System	12-13
7	Coding Challenge & Leaderboard Platform	14-15
8	Career Networking & Mentorship Platform	16-17
9	AI-Based Personalized Shopping Recommendation System	18-19
10	AI-Powered Discussion Forum	20-21

1. Smart Drone Delivery Management System

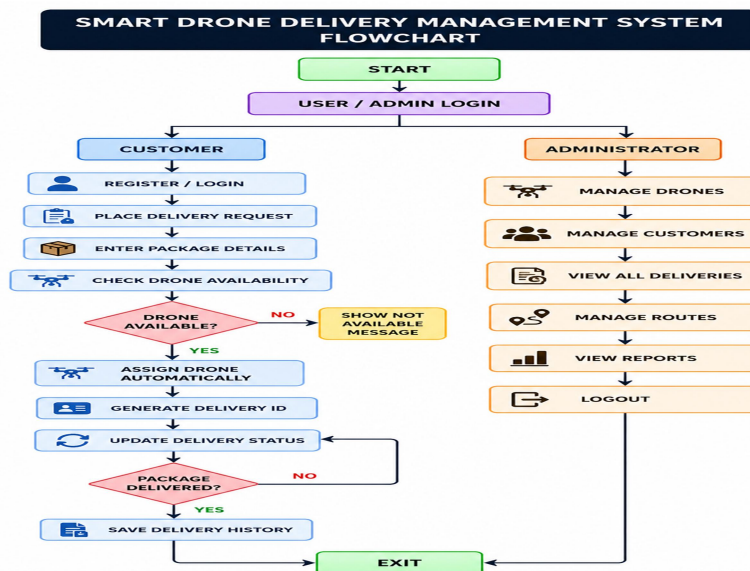
Project Description

Smart Drone Delivery Management System is a Java-based application designed to automate and manage drone-based package delivery services. The system enables customers to place delivery requests, while administrators manage drones, delivery routes, packages, and delivery status. It simulates how modern logistics companies use autonomous drones for fast and efficient last-mile delivery. The application is developed using Java with Object-Oriented Programming (OOP) concepts such as Encapsulation, Inheritance, Polymorphism, and Abstraction. Java Collections like ArrayList, HashMap, and Queue are used to manage customer details, drone information, package records, and delivery requests efficiently.

The system provides modules for customer registration, package booking, drone allocation, route management, delivery tracking, and delivery history. Administrators can add drones, update drone availability, monitor battery status, and assign deliveries based on package weight and drone capacity. Exception handling ensures smooth operation by preventing invalid inputs and scheduling conflicts.

The application can be implemented as a Java Swing GUI or a console-based application, with optional MySQL/JDBC integration for permanent data storage. Customers can track the delivery status in real time (simulated), while administrators monitor all ongoing deliveries through a dashboard. This project introduces students to modern logistics automation, smart transportation systems, resource scheduling, and data management using Java. It also provides practical experience in designing enterprise-level software for future delivery technologies.

Flowchart Diagram



Why this Project?

- Drone delivery is one of the fastest-growing technologies in logistics and e-commerce.
- Demonstrates automation in transportation and package management.
- Introduces resource allocation and scheduling concepts.
- Provides practical experience in Java GUI development, Collections, and OOP.
- Simulates real-world smart delivery systems used by companies worldwide.

Key Features

- Customer Registration and Login
- Drone Registration and Management
- Package Booking
- Automatic Drone Assignment
- Delivery Route Management
- Package Tracking (Simulation)
- Drone Battery Status Monitoring
- Delivery Status Updates
- Delivery History

Expected Outcome

- A working simulation of a smart drone delivery platform.
- Understanding of logistics automation and scheduling.
- Experience with Java-based enterprise application development.
- Improved knowledge of OOP, Collections Framework, and GUI programming.
- Practical exposure to smart transportation and delivery management systems.

2. Digital Wallet & UPI Payment Simulator

Project Description

Digital Wallet & UPI Payment Simulator is a Java-based application that simulates the functionality of modern digital payment platforms such as Google Pay, PhonePe, Paytm, and BHIM UPI. The system allows users to create digital wallets, register bank accounts, add money, transfer funds using UPI IDs, pay bills, and view transaction history in a secure environment.

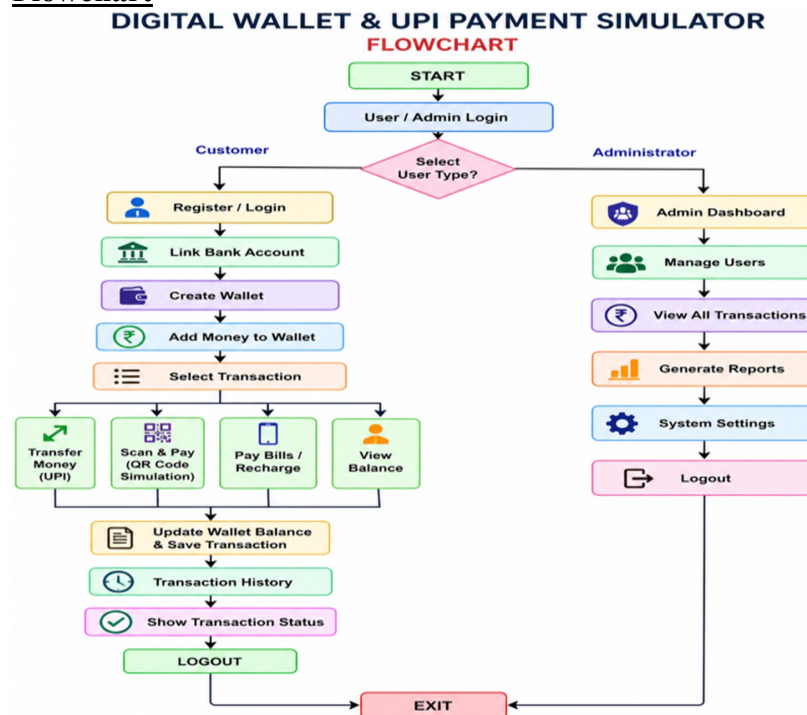
The application is developed using Java and Object-Oriented Programming (OOP) concepts such as Encapsulation, Inheritance, Polymorphism, and Abstraction. Java Collections like ArrayList, HashMap, and LinkedList are used to manage user accounts, wallet balances, bank details, UPI IDs, and transaction records efficiently.

The system consists of modules for user registration, wallet management, bank account linking, UPI fund transfers, bill payments, QR code payment simulation, and transaction history. Administrators can monitor users, verify transactions, generate reports, and manage the overall payment system.

The application can be implemented using Java Swing for the graphical interface or as a console-based application, with optional MySQL/JDBC integration for permanent data storage. Validation and exception handling ensure secure and error-free transactions.

This project introduces students to the working principles of digital payment systems, online banking, and secure financial transaction processing while strengthening Java programming skills.

Flowchart



Why this Project?

- Digital payments are one of the fastest-growing technologies worldwide.
- Demonstrates secure online financial transaction processing.
- Introduces students to UPI payment architecture.
- Enhances Java programming using Collections, OOP, Swing, and JDBC.
- Simulates real-world payment applications.

Key Features

- User Registration & Login
- Digital Wallet Creation
- Bank Account Linking
- UPI ID Generation
- Add Money to Wallet
- Money Transfer
- QR Code Payment (Simulation)
- Mobile Recharge
- Electricity Bill Payment
- Transaction History
- Wallet Balance
- Admin Dashboard
- Transaction Reports

Expected Outcome

- Working Digital Wallet Simulation.
- Secure UPI Money Transfer Simulation.
- Experience in Java GUI development.
- Knowledge of transaction management.
- Understanding of digital banking concepts.
- Practical implementation of Collections and JDBC.

3. Smart Campus Event Management Portal

Project Description

Smart Campus Event Management Portal is a Java-based application designed to simplify the planning, registration, and management of college events such as workshops, seminars, hackathons, technical fests, cultural programs, sports events, and guest lectures. The system provides a centralized platform where students can register for events online while organizers can efficiently manage schedules, venues, participants, and event reports.

The application is developed using Java, following Object-Oriented Programming (OOP) principles including Encapsulation, Inheritance, Polymorphism, and Abstraction.

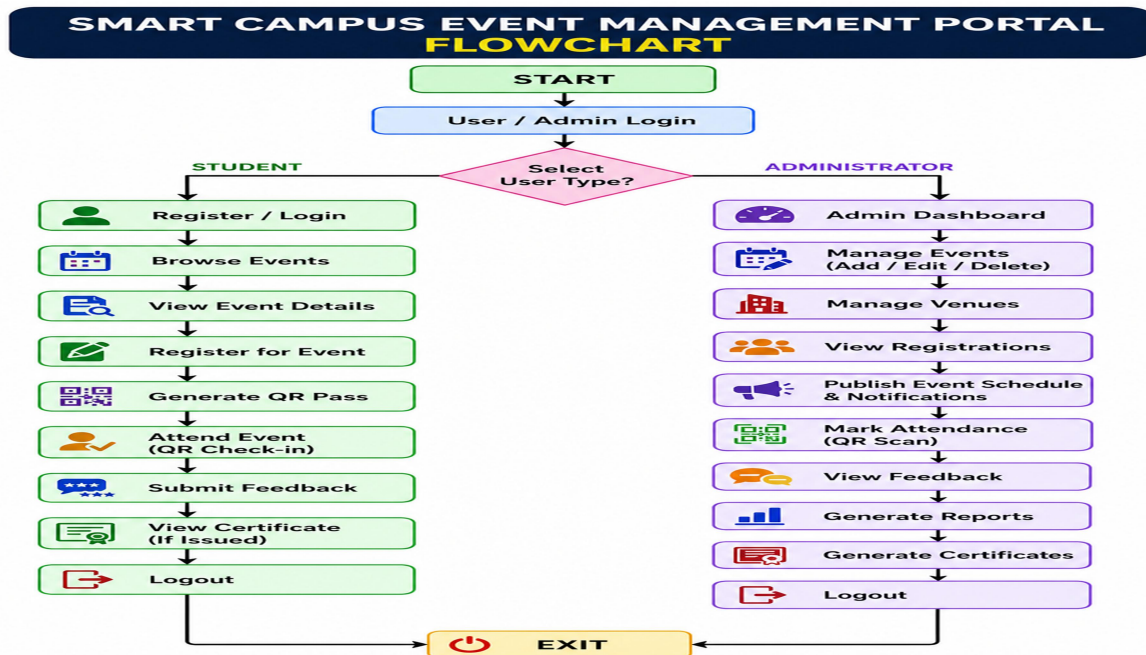
Java Collections such as ArrayList, HashMap, and PriorityQueue are used to store and manage event information, student registrations, venue details, and participant lists efficiently.

The system includes modules for user authentication, event creation, online registration, QR-based attendance simulation, notification management, certificate generation, and feedback collection. Event coordinators can monitor registrations, approve participants, assign venues, publish event schedules, and generate participation reports.

The application can be implemented using Java Swing for the graphical interface or as a console-based application. JDBC with MySQL or File Handling can be used for permanent data storage. Proper validation and exception handling ensure reliable and secure operation.

This project introduces students to modern event automation systems and provides practical experience in Java application development, database connectivity, and software project design.

Flowchart



Why this Project?

- Automates college event management.
- Eliminates manual registration and paperwork.
- Provides a centralized platform for organizers and students.
- Demonstrates Java Swing, Collections, JDBC, and OOP concepts.
- Simulates modern campus management systems.

Key Features

- Student Registration & Login
- Admin Login
- Event Creation
- Event Registration
- QR Code Pass Generation (Simulation)
- Venue Management
- Event Schedule Management
- Participant Management
- Certificate Generation
- Feedback Collection
- Event Notifications
- Reports & Analytics

Expected Outcome

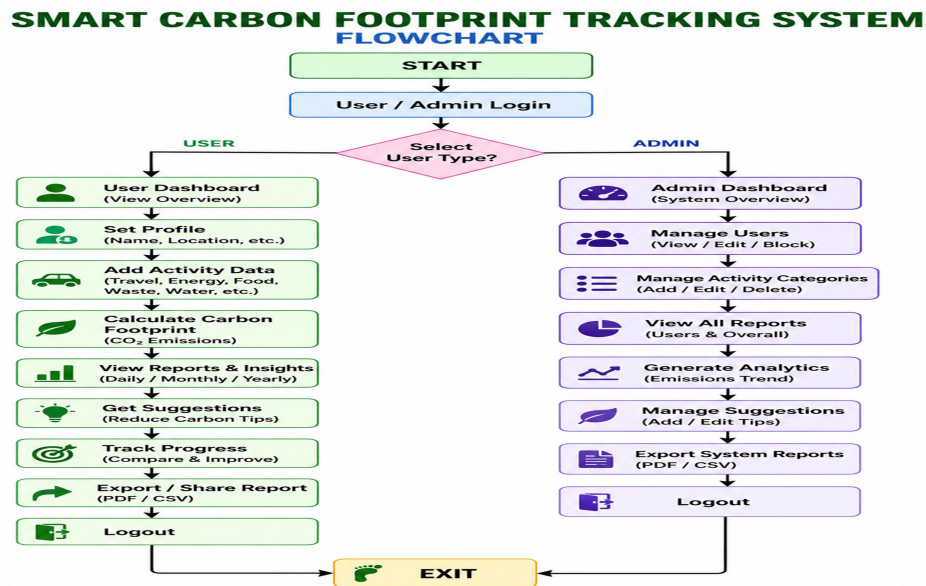
- A complete digital campus event management system.
- Automated event registration and attendance.
- Efficient venue and schedule management.
- Practical experience in Java GUI and database programming.
- Understanding of event workflow automation.

4. Smart Carbon Footprint Tracking System

Project Description

Smart Carbon Footprint Tracking System is a Java-based application that helps individuals and organizations monitor, calculate, and reduce their carbon emissions. The system collects user activities such as transportation, electricity usage, water consumption, fuel usage, and waste generation to estimate the total carbon footprint. Based on the calculated emissions, the application provides eco-friendly suggestions to reduce environmental impact. The application is developed using **Java** with **Object-Oriented Programming (OOP)** concepts including Encapsulation, Inheritance, Polymorphism, and Abstraction. Java Collections such as **ArrayList**, **HashMap**, and **LinkedList** are used to store user profiles, activity records, emission data, and reports efficiently. The system allows users to register, log in, record daily activities, calculate carbon emissions, view reports, and monitor their progress over time. Administrators can manage users, update emission factors, generate environmental reports, and analyze overall system statistics. The project can be implemented using **Java Swing** for the graphical user interface or as a **console-based application**, with optional **MySQL/JDBC** integration for permanent data storage. This project provides students with practical experience in Java programming while promoting awareness of environmental sustainability and green technology.

Flowchart



Why this Project?

- Encourages environmental awareness.
- Demonstrates real-world data processing.
- Uses Java OOP, Collections, Swing, and JDBC.
- Helps understand sustainability and carbon emission concepts.
- Suitable for smart city and green technology applications.

Key Features

- User Registration & Login
- Daily Activity Tracking
- Electricity Consumption Monitoring
- Fuel Usage Tracking
- Vehicle Travel Calculation
- Water Usage Monitoring
- Waste Generation Tracking
- Carbon Footprint Calculation
- Monthly Reports
- Carbon Reduction Suggestions
- Progress Dashboard
- Administrator Dashboard

Expected Outcome

- Calculate users' carbon emissions accurately.
- Generate environmental impact reports.
- Help users adopt eco-friendly habits.
- Improve Java programming skills.
- Understand sustainability and green computing concepts.

5. Disaster Alert & Emergency Response System

Project Description

Disaster Alert & Emergency Response System is a Java-based application designed to provide early disaster alerts and coordinate emergency response activities during natural and man-made disasters such as floods, earthquakes, cyclones, fires, and landslides. The system enables users to report emergencies, receive disaster alerts, locate nearby shelters, and request emergency assistance through a centralized platform.

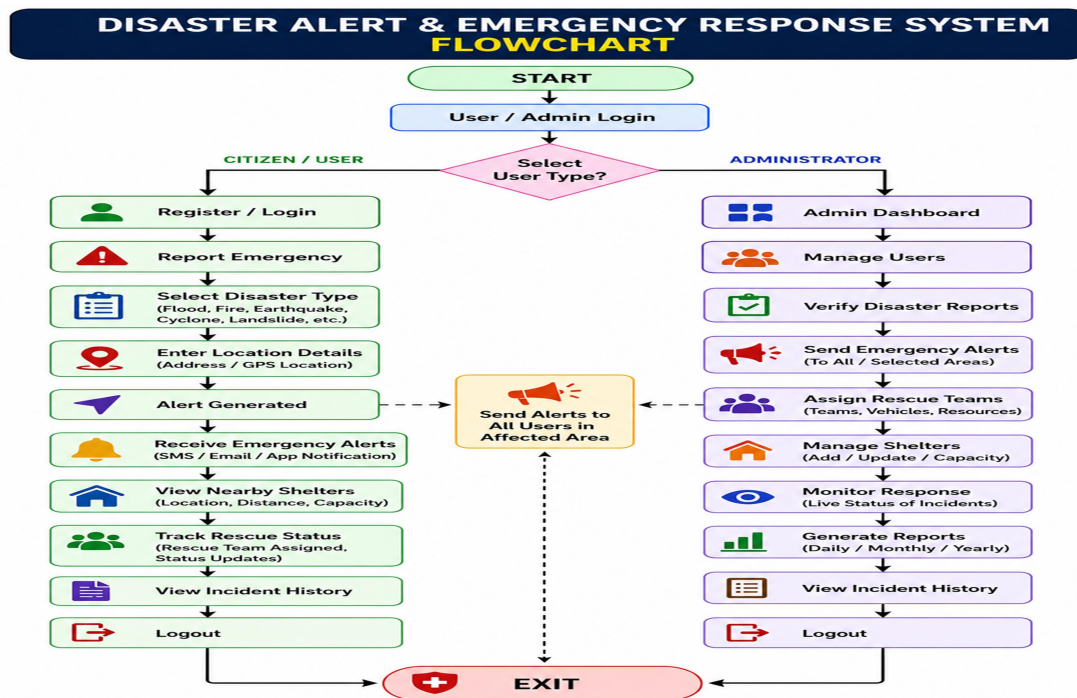
The application is developed using **Java** and follows **Object-Oriented Programming (OOP)** principles such as Encapsulation, Inheritance, Polymorphism, and Abstraction. Java Collections including **ArrayList**, **HashMap**, and **Queue** are used to manage disaster reports, emergency contacts, rescue teams, shelter information, and alert notifications efficiently.

The system consists of modules for user registration, disaster reporting, emergency alert broadcasting, shelter management, rescue team coordination, and report generation. Administrators can monitor incidents, verify disaster reports, assign rescue teams, update emergency notifications, and generate disaster management reports.

The application can be implemented using **Java Swing** for the graphical user interface or as a **console-based application**, with optional **MySQL/JDBC** integration for permanent data storage.

This project helps students understand disaster management systems, emergency communication, and resource allocation while improving their Java programming, GUI development, and database skills.

Flowchart



Why this Project?

- Helps improve disaster preparedness and emergency response.
- Demonstrates real-world emergency management systems.
- Enhances Java programming using Swing, Collections, and JDBC.
- Introduces resource allocation and notification systems.
- Supports smart city and public safety initiatives

Key Features

- User Registration & Login
- Disaster Reporting
- Emergency Alert Notification
- Disaster Type Selection
- Shelter Locator
- Rescue Team Assignment
- Emergency Contact Directory
- Live Status Tracking (Simulation)
- Incident History
- Administrator Dashboard
- Disaster Analytics & Reports

Expected Outcome

- A functional disaster management application.
- Quick emergency reporting and alert simulation.
- Efficient rescue team and shelter management.
- Practical understanding of Java Swing and JDBC.
- Awareness of disaster response systems and emergency planning

6. EV Charging Station Booking System

Project Description

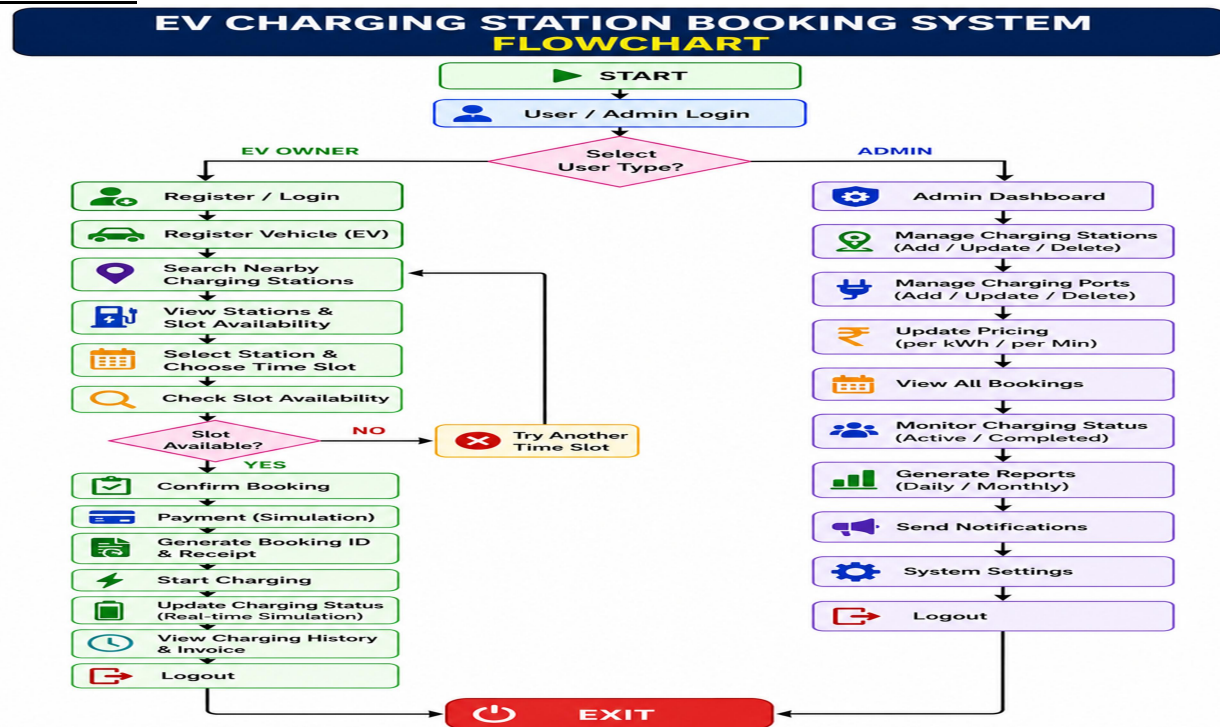
EV Charging Station Booking System is a Java-based application designed to help electric vehicle (EV) owners locate, book, and manage charging stations efficiently. As the adoption of electric vehicles continues to grow, charging stations often become busy, making advance booking essential. This system allows users to reserve charging slots, monitor charging status, and make online payments (simulation), while administrators manage charging stations, charging ports, bookings, and usage reports.

The application is developed using **Java** and follows **Object-Oriented Programming (OOP)** principles such as Encapsulation, Inheritance, Polymorphism, and Abstraction. Java Collections like **ArrayList**, **HashMap**, and **PriorityQueue** are used to manage user accounts, charging stations, booking records, charging slots, and payment details efficiently.

The system provides modules for user registration, EV registration, charging station search, slot booking, payment simulation, charging history, and notifications. Administrators can add charging stations, monitor charging port availability, update pricing, manage bookings, and generate daily usage reports.

The project can be implemented using **Java Swing** for the graphical interface or as a **console-based application**, with optional **MySQL/JDBC** integration for permanent data storage. This project gives students practical experience in Java application development while introducing concepts related to smart transportation, renewable energy, and electric vehicle infrastructure.

Flowchart



Why this Project?

- Supports the growing adoption of electric vehicles.
- Simulates smart charging station management.
- Demonstrates Java GUI, Collections, OOP, and JDBC concepts.
- Introduces booking systems and scheduling algorithms.
- Promotes green energy and sustainable transportation.

Key Features

- User Registration & Login
- EV Registration
- Search Charging Stations
- View Charging Port Availability
- Slot Booking
- Online Payment Simulation
- Booking Cancellation
- Charging Status Tracking
- Charging History
- Notifications
- Admin Dashboard
- Reports & Analytics

Expected Outcome

- A working EV charging station booking application.
- Efficient charging slot scheduling.
- Improved understanding of Java Swing and JDBC.
- Practical experience in booking and payment systems.
- Knowledge of EV infrastructure management.

7. Coding Challenge & Leaderboard Platform

Project Description

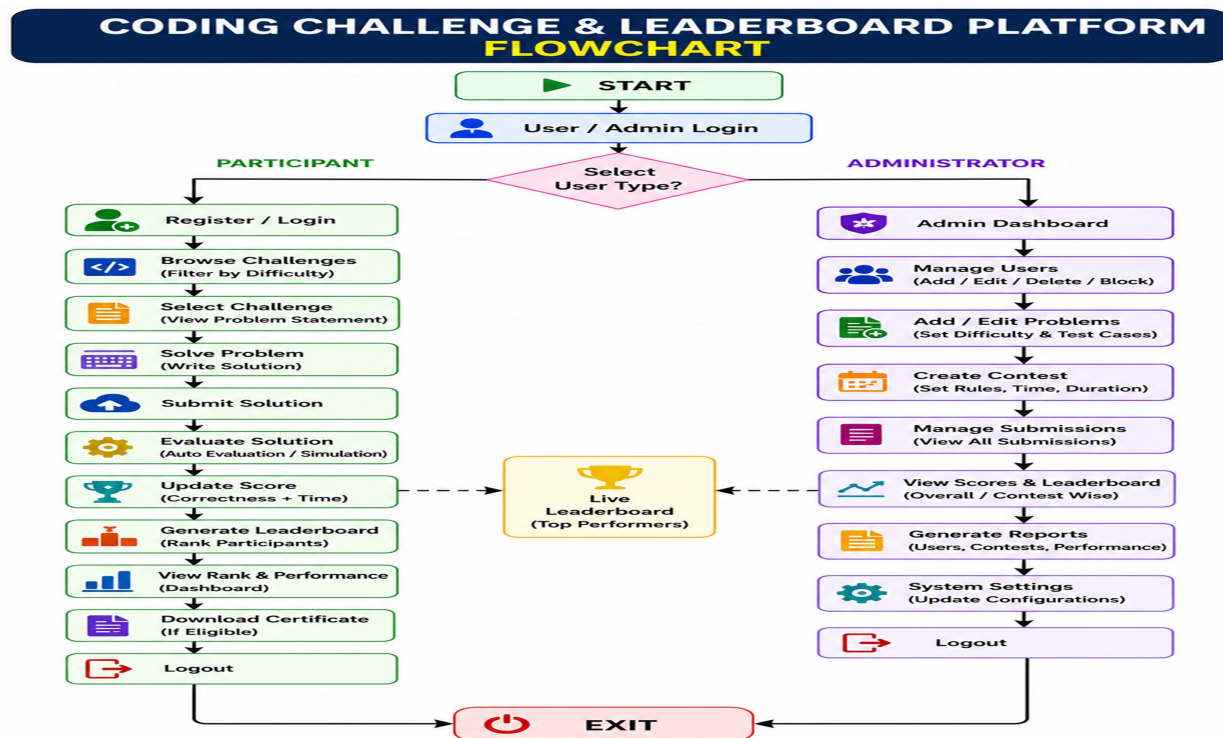
Coding Challenge & Leaderboard Platform is a Java-based application that allows students and programmers to practice coding problems, participate in programming contests, and track their performance through a dynamic leaderboard. The platform provides coding challenges categorized by difficulty levels, evaluates submitted solutions (simulation), awards scores, and ranks participants based on their performance.

The application is developed using Java and follows Object-Oriented Programming (OOP) principles such as Encapsulation, Inheritance, Polymorphism, and Abstraction. Java Collections like ArrayList, HashMap, PriorityQueue, and TreeMap are used to manage users, coding problems, submissions, scores, and leaderboard rankings efficiently.

The system includes modules for user registration, coding challenge management, contest participation, code submission (simulation), automatic score calculation, leaderboard generation, and performance analytics. Administrators can create contests, add coding problems, monitor submissions, manage users, and generate reports.

The project can be implemented using Java Swing for the graphical user interface or as a console-based application, with optional MySQL/JDBC integration for permanent data storage. This project provides students with practical experience in Java programming, competitive coding platforms, data structures, and software development.

Flowchart



Why this Project?

- Simulates platforms like LeetCode, HackerRank, and CodeChef.
- Encourages competitive programming and coding skills.
- Demonstrates Java Collections, OOP, Swing, and JDBC.
- Provides hands-on experience with ranking algorithms.
- Suitable for coding clubs and programming contests.

Key Features

- User Registration & Login
- Coding Challenge Repository
- Difficulty Levels (Easy, Medium, Hard)
- Contest Management
- Code Submission (Simulation)
- Automatic Score Calculation
- Live Leaderboard
- User Ranking
- Performance Dashboard
- Certificate Generation
- Admin Dashboard
- Reports & Analytics

Expected Outcome

- A fully functional coding practice platform.
- Automatic score calculation and ranking.
- Contest management with leaderboard generation.
- Improved understanding of Java Collections, OOP, Swing, and JDBC.
- Practical exposure to competitive programming systems.

8. Career Networking & Mentorship Platform

Project Description

Career Networking & Mentorship Platform is a Java-based application that connects students, alumni, mentors, and recruiters through a single digital platform. The system helps students build professional networks, seek career guidance, schedule mentorship sessions, explore internships and job opportunities, and improve their career development. Mentors can share career advice, conduct mock interviews, recommend learning resources, and monitor mentee progress.

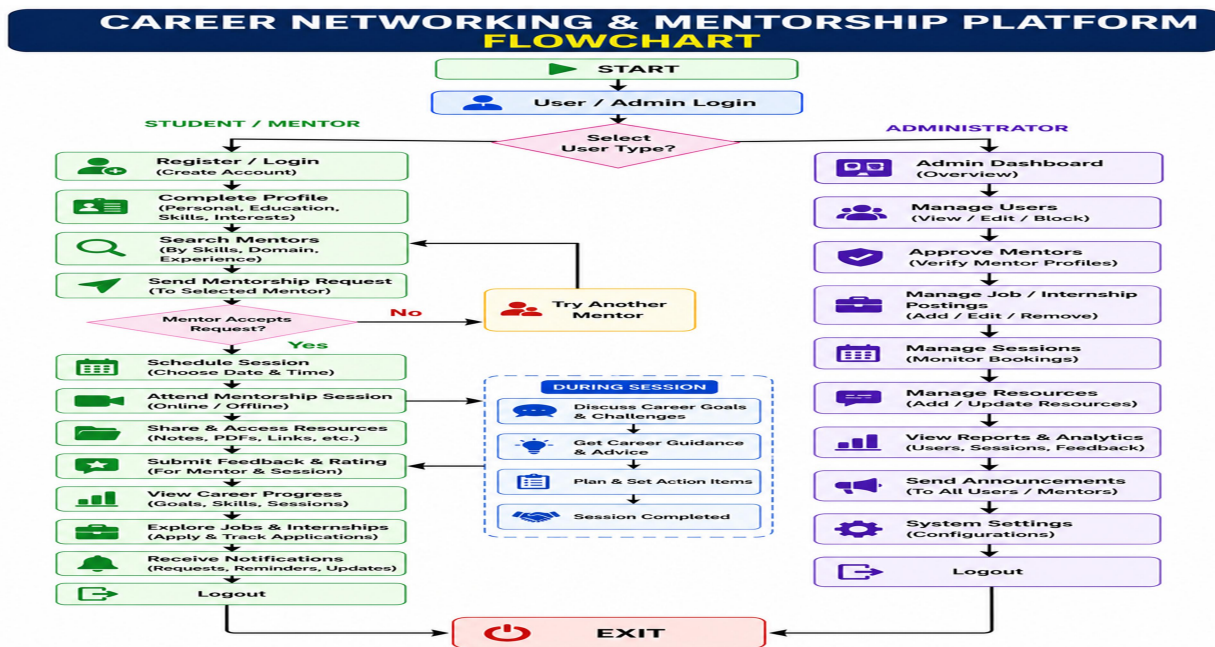
The application is developed using **Java** and follows **Object-Oriented Programming (OOP)** concepts such as Encapsulation, Inheritance, Polymorphism, and Abstraction. Java Collections including **ArrayList**, **HashMap**, and **PriorityQueue** are used to manage users, mentor profiles, mentorship requests, appointments, and career resources efficiently.

The system consists of modules for user registration, mentor discovery, mentorship booking, messaging, career resource management, internship and job postings, feedback collection, and report generation. Administrators can manage users, approve mentors, monitor activities, publish announcements, and generate platform analytics.

The application can be implemented using **Java Swing** for the graphical user interface or as a **console-based application**, with optional **MySQL/JDBC** integration for permanent data storage.

This project provides students with practical experience in Java application development while addressing a real-world need for career guidance and professional networking.

Flowchart



Why this Project?

- Promotes career development and professional networking.
- Connects students with alumni and industry mentors.
- Demonstrates Java Swing, Collections, OOP, and JDBC concepts.
- Simulates modern mentoring platforms used by universities and companies.
- Enhances communication and career planning skills.

Key Features

- Student & Mentor Registration
- Secure Login
- Profile Management
- Mentor Search
- Mentorship Request
- Session Scheduling
- Career Guidance
- Internship & Job Listings
- Resource Sharing
- Feedback & Ratings
- Notifications
- Admin Dashboard

Expected Outcome

- A complete career networking and mentorship portal.
- Efficient mentor matching and session scheduling.
- Organized internship and job management.
- Improved Java programming, GUI, and database skills.
- Better understanding of professional networking systems.

9. AI-Based Personalized Shopping Recommendation System

Project Description

AI-Based Personalized Shopping Recommendation System is a Java-based application that helps users discover products based on their interests, browsing history, purchase history, ratings, and preferences. The system simulates an AI recommendation engine by analyzing user behavior and suggesting products that best match individual preferences.

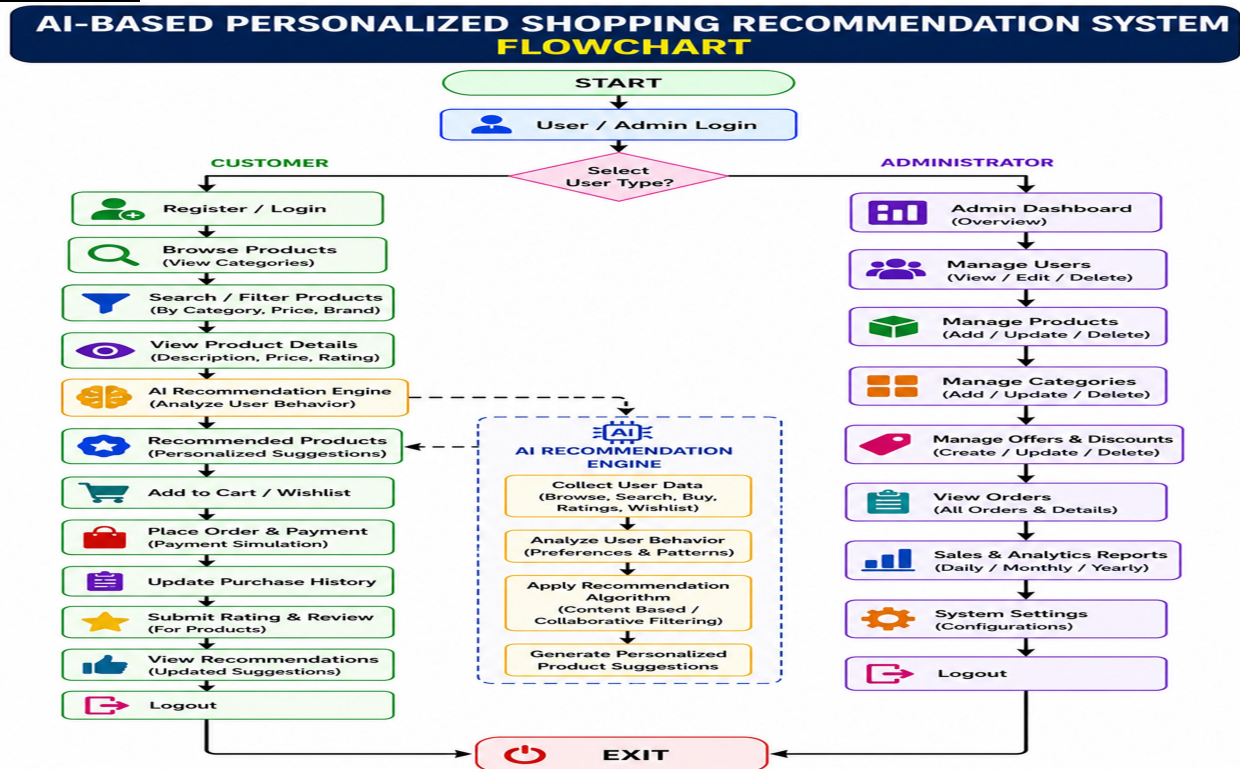
The application is developed using **Java** and follows **Object-Oriented Programming (OOP)** concepts such as Encapsulation, Inheritance, Polymorphism, and Abstraction. Java Collections like **ArrayList**, **HashMap**, **PriorityQueue**, and **TreeMap** are used to manage users, products, shopping history, recommendations, and ratings efficiently.

The system includes modules for user registration, product browsing, shopping cart management, order processing, personalized recommendations, wishlist management, product ratings, and purchase history. Administrators can manage products, categories, users, offers, and sales reports.

The application can be implemented using **Java Swing** for the graphical user interface or as a **console-based application**, with optional **MySQL/JDBC** integration for permanent data storage.

This project introduces students to recommendation systems, customer analytics, and e-commerce technologies while strengthening Java programming, GUI development, and database management skills.

Flowchart



Why this Project?

- Simulates AI-powered recommendation systems used by major e-commerce platforms.
- Demonstrates customer personalization and recommendation logic.
- Enhances Java programming using Swing, Collections, OOP, and JDBC.
- Introduces students to recommendation algorithms and e-commerce applications.
- Develops practical software engineering skills.

Key Features

- User Registration & Login
- Product Catalog
- Product Search & Filters
- AI-Based Product Recommendations (Simulation)
- Shopping Cart
- Wishlist
- Order Management
- Payment Simulation
- Product Ratings & Reviews
- Purchase History
- Admin Dashboard
- Sales Reports

Expected Outcome

- A personalized shopping recommendation application.
- AI-style product suggestions based on user activity.
- Efficient shopping cart and order management.
- Practical experience in Java GUI, Collections, and JDBC.
- Understanding of recommendation system concepts used in modern e-commerce.

10. AI-Powered Discussion Forum

Project Description

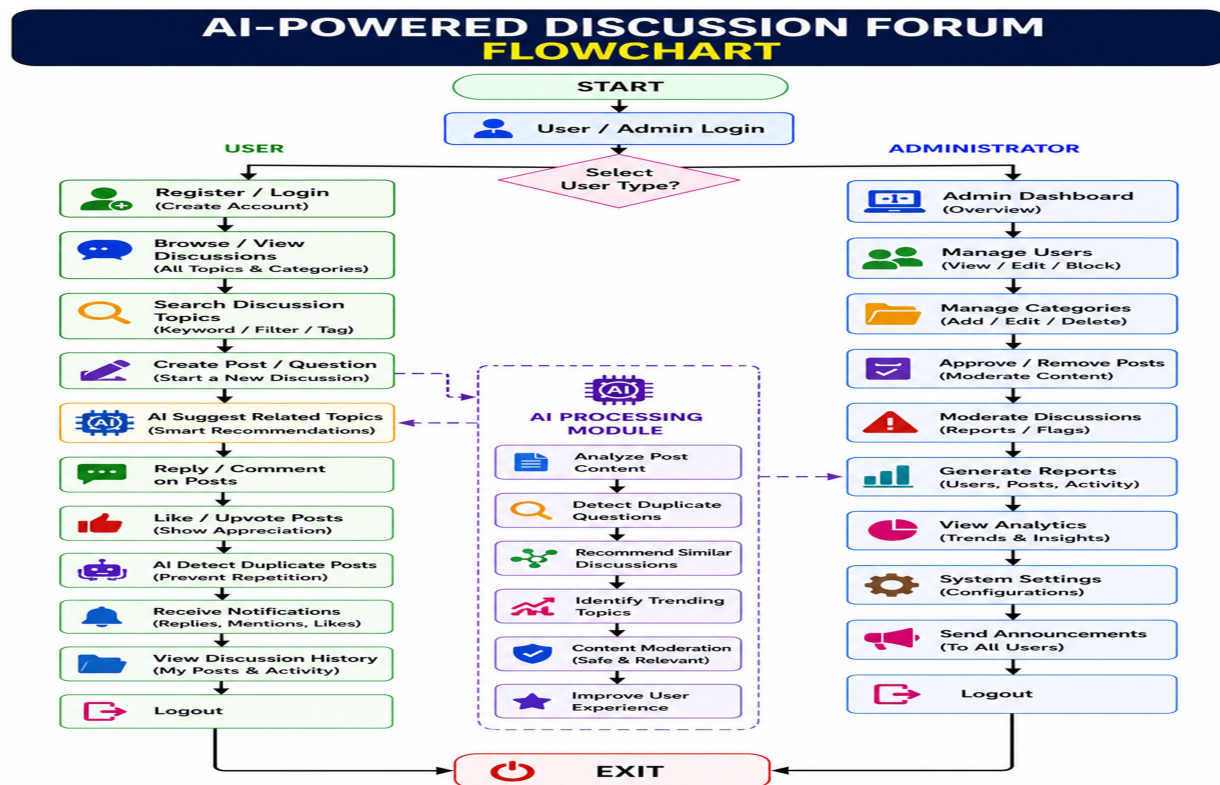
AI-Powered Discussion Forum is a Java-based application that enables students, faculty, and professionals to participate in intelligent online discussions. Users can create discussion topics, post questions, share answers, comment on posts, and receive AI-powered suggestions for related discussions and answers. The system also uses AI simulation to categorize posts, detect duplicate questions, recommend relevant content, and moderate inappropriate comments.

The application is developed using **Java** and follows **Object-Oriented Programming (OOP)** principles such as **Encapsulation, Inheritance, Polymorphism, and Abstraction**. Java Collections including **ArrayList, HashMap, PriorityQueue, and TreeMap** are used to manage users, discussion topics, posts, comments, likes, and AI recommendations efficiently.

The platform consists of modules for user registration, discussion management, AI-based content recommendation, notifications, moderation, and analytics. Administrators can manage users, approve or remove posts, monitor discussions, generate reports, and configure forum settings. The application can be implemented using **Java Swing** for the graphical user interface or as a **console-based application**, with optional **MySQL/JDBC** integration for permanent data storage.

This project introduces students to online collaboration platforms, recommendation systems, AI-assisted moderation, and Java application development.

Flowchart



Why this Project?

- Encourages collaborative learning and knowledge sharing.
- Simulates AI-powered recommendation systems used in modern forums.
- Demonstrates Java Swing, OOP, Collections, and JDBC concepts.
- Helps students understand intelligent content management.
- Suitable for college discussion portals and technical communities.

Key Features

- User Registration & Login
- Discussion Categories
- Create Questions & Posts
- AI-Based Topic Recommendation (Simulation)
- Duplicate Question Detection
- Comment & Reply System
- Like & Upvote System
- Search Discussions
- Notifications
- User Reputation Score
- Admin Dashboard
- Reports & Analytics

Expected Outcome

- A functional AI-assisted discussion platform.
- Organized discussion and knowledge sharing.
- Intelligent content recommendations.
- Practical experience in Java GUI, Collections, and JDBC.
- Understanding of collaborative platforms and AI-based moderation.